

Міністерство освіти і науки України  
Національний технічний університет України «КПІ» імені Ігоря Сікорського  
Кафедра обчислювальної техніки ФІОТ

ЗВІТ  
з лабораторної роботи №3  
з навчальної дисципліни «Методи наукових досліджень»

Тема:  
ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ  
ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ

Виконав:  
Студент групи ІВ-92,  
Карпека Дмитрій Юрійович

Перевірив:  
Регіда П. Г.

Київ 2021

**Мета:** провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

### Завдання на лабораторну роботу

1. Скласти матрицю планування для дробового трьохфакторного експерименту. Провести експеримент в усіх точках факторного простору, повторивши N експериментів, де N – кількість експериментів (рядків матриці планування) в усіх точках факторного простору – знайти значення функції відгуку Y. Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі (випадковим чином).

$$y_{\max} = 200 + x_{\text{ср max}};$$

$$y_{\min} = 200 + x_{\text{ср min}}$$

$$\text{де } x_{\text{ср max}} = \frac{x_{1\max} + x_{2\max} + x_{3\max}}{3}, \quad x_{\text{ср min}} = \frac{x_{1\min} + x_{2\min} + x_{3\min}}{3}$$

2. Знайти коефіцієнти лінійного рівняння регресії. Записати лінійне рівняння регресії.

3. Провести 3 статистичні перевірки.

4. Написати комп'ютерну програму, яка усе це виконує.

Варіант:

210	-25	-5	-70	-10	-25	-5
-----	-----	----	-----	-----	-----	----

Виконання роботи:

1) Результати роботи програми:

3) Заповнена матриця планування експерименту для m=3:

```
+-----+-----+-----+-----+
| X0 | X1 | X2 | X3 | Y1 | Y2 | Y3 |
+-----+-----+-----+-----+
| 1  | -1 | -1 | -1 | 183 | 174 | 181 |
| 1  | -1 | 1  | 1  | 164 | 179 | 192 |
| 1  | 1  | -1 | 1  | 191 | 171 | 168 |
| 1  | 1  | 1  | -1 | 191 | 171 | 168 |
+-----+-----+-----+-----+
```

Значення факторів нормовані

Середнє значення ф-ції відгуку у рядку:

y1 = 179.33

y2 = 178.33

y3 = 176.67

y4 = 176.67

Отримане рівняння регресії:

$y = 175.422 + (-0.108)*x1 + (-0.008)*x2 + (-0.025)*x3$

Перевірка:

$y0 = 175.422 + (-0.108)*-25 + (-0.008)*-70 + (-0.025)*-25 = 179.33$

$y1 = 175.422 + (-0.108)*-25 + (-0.008)*-10 + (-0.025)*-5 = 178.33$

$y2 = 175.422 + (-0.108)*-5 + (-0.008)*-70 + (-0.025)*-5 = 176.67$

$y3 = 175.422 + (-0.108)*-5 + (-0.008)*-10 + (-0.025)*-25 = 176.67$

Дисперсія однорідна

$\sigma_p = 0.37 < \sigma_t = 0.906$

Коефіцієнти  $[-0.108, -0.008, -0.025]$  приймаються незначними при рівні значимості 0.05, тобто вони виключаються із рівняння

Коефіцієнти, що залишились [175.422]

Значення 'y' із коефіцієнтами [175.422]:

y1 = 175.422

y2 = 175.422

y3 = 175.422

$F_p = 3.637571799551866 < F_t = 4.06618055135116$ , отже математична модель адекватна експериментальним даним

Код програми:

```
from random import randint
import prettytable
import math
import numpy as np
from scipy.stats import f, t
from functools import partial
import math
```

```
# Лабораторна робота №3 "ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ
# ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ" з предмету МОПЕ
# Варіант №210 Карпека Дмитрій
```

```
#-----Початкові умови-----
-----
```

```
variant = 210
```

```
#-----1-----
-----
```

```
# y = b0 + b1x1 + b2x2
```

```
#-----2-----
-----
```

```
x0 = 1
```

```
x1_min = -25
```

```
x1_max = -5
```

```
x2_min = -70
```

```
x2_max = -10
```

```
x3_min = -25
```

```

x3_max = -5

x1_min_norm, x2_min_norm, x3_min_norm = -1, -1, -1
x1_max_norm, x2_max_norm, x3_max_norm = 1, 1, 1

m = 3
n = 4

x_average_max = sum([x1_max, x2_max, x3_max])/3
x_average_min = sum([x1_min, x2_min, x3_min])/3

y_max = round(200 + x_average_max)
y_min = round(200 + x_average_min)

pt = prettytable.PrettyTable()
pt.field_names = ["X0", "X1", "X2", "X3"] + ["Y" + str(x) for x in range(1,
m+1)]

#-----3-----
def matrix_plan(m, ymin, ymax, n=n):
    return [[randint(ymin, ymax) for _ in range(m)] for _ in range(n)]

matrix_y = matrix_plan(m, y_min, y_max)
matrix_y = [[x0, x1_min_norm, x2_min_norm, x3_min_norm] + matrix_y[0],
            [x0, x1_min_norm, x2_max_norm, x3_max_norm] + matrix_y[1],
            [x0, x1_max_norm, x2_min_norm, x3_max_norm] + matrix_y[2],
            [x0, x1_max_norm, x2_max_norm, x3_min_norm] + matrix_y[2]]
pt.add_rows(matrix_y)

print("3) Заповнена матриця планування експерименту для m={0}:".format(m))
print(pt)
print("Значення факторів нормовані")

#-----4-----
#4.1
def average(list):
    return sum(list) / len(list)

y1 = round(average(matrix_y[0][m+1:len(matrix_y[0])]), 2)
y2 = round(average(matrix_y[1][m+1:len(matrix_y[1])]), 2)
y3 = round(average(matrix_y[2][m+1:len(matrix_y[2])]), 2)
y4 = round(average(matrix_y[3][m+1:len(matrix_y[3])]), 2)
print("Середнє значення ф-ції відгуку у рядку:")
for _ in range(0, m+1):
    yn = round(average(matrix_y[_][m+1:len(matrix_y[_])]), 2)
    print("Y{0} = {1}".format(_+1, yn))

matrix_natur = [[x1_min, x2_min, x3_min],
                [x1_min, x2_max, x3_max],
                [x1_max, x2_min, x3_max],
                [x1_max, x2_max, x3_min]]
# TODO: mxn, my, an DONE
mx1 = average([i[0] for i in matrix_natur])

```

```

mx2 = average([i[1] for i in matrix_natur])
mx3 = average([i[2] for i in matrix_natur])
my = average([y1, y2, y3, y4])

a1 = average([matrix_natur[0][0]*y1, matrix_natur[1][0]*y2,
matrix_natur[2][0]*y3, matrix_natur[3][0]*y4])
a2 = average([matrix_natur[0][1]*y1, matrix_natur[1][1]*y2,
matrix_natur[2][1]*y3, matrix_natur[3][1]*y4])
a3 = average([matrix_natur[0][2]*y1, matrix_natur[1][2]*y2,
matrix_natur[2][2]*y3, matrix_natur[3][2]*y4])
# TODO: aij, i==j DONE
a11 = average([x*x for x in [matrix_natur[0][0], matrix_natur[1][0],
matrix_natur[2][0], matrix_natur[3][0]]])
a22 = average([x*x for x in [matrix_natur[0][1], matrix_natur[1][1],
matrix_natur[2][1], matrix_natur[3][1]]])
a33 = average([x*x for x in [matrix_natur[0][2], matrix_natur[1][2],
matrix_natur[2][2], matrix_natur[3][2]]])
# TODO: aij DONE
a12 = average([matrix_natur[0][0]*matrix_natur[0][1],
matrix_natur[1][0]*matrix_natur[1][1], matrix_natur[2][0]*matrix_natur[2][1],
matrix_natur[3][0]*matrix_natur[3][1]])
a21 = a12
a13 = average([matrix_natur[0][0]*matrix_natur[0][2],
matrix_natur[1][0]*matrix_natur[1][2], matrix_natur[2][0]*matrix_natur[2][2],
matrix_natur[3][0]*matrix_natur[3][2]])
a31 = a13
a23 = average([matrix_natur[0][1]*matrix_natur[0][2],
matrix_natur[1][1]*matrix_natur[1][2], matrix_natur[2][1]*matrix_natur[2][2],
matrix_natur[3][1]*matrix_natur[3][2]])
a32 = a23
# TODO: bi DONE
b0 = np.linalg.det(np.array([my, mx1, mx2, mx3], [a1, a11, a12, a13], [a2, a12,
a22, a32], [a3, a13, a23, a33]))) / np.linalg.det(np.array([[1, mx1, mx2, mx3],
[mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]))
b1 = np.linalg.det(np.array([[1, my, mx2, mx3], [mx1, a1, a12, a13], [mx2, a2,
a22, a32], [mx3, a3, a23, a33]])) / np.linalg.det(np.array([[1, mx1, mx2, mx3],
[mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]))
b2 = np.linalg.det(np.array([[1, mx1, my, mx3], [mx1, a11, a1, a13], [mx2, a12,
a2, a32], [mx3, a13, a3, a33]])) / np.linalg.det(np.array([[1, mx1, mx2, mx3],
[mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]))
b3 = np.linalg.det(np.array([[1, mx1, mx2, my], [mx1, a11, a12, a1], [mx2, a12,
a22, a2], [mx3, a13, a23, a3]])) / np.linalg.det(np.array([[1, mx1, mx2, mx3],
[mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]))
print("Отримане рівняння регресії:\ny = {0} + ({1})*x1 + ({2})*x2 +
({3})*x3".format(round(b0, 3), round(b1, 3), round(b2, 3), round(b3, 3)))
# TODO: check DONE
print("Перевірка:")
for i in range(4):
    y = b0 + b1*matrix_natur[i][0] + b2*matrix_natur[i][1] +
b3*matrix_natur[i][2]
    print("y{0} = {1} + ({2})*{3} + ({4})*{5} + ({6})*{7} = {8}".format(i,
round(b0,3), round(b1,3), matrix_natur[i][0], round(b2,3), matrix_natur[i][1],
round(b3,3), matrix_natur[i][2], round(y,2)))

# TODO Cohren criteria DONE
s_sq_y1 = average([(y1j - y1)**2 for y1j in matrix_y[0][m+1:len(matrix_y[0])]))
s_sq_y2 = average([(y2j - y2)**2 for y2j in matrix_y[1][m+1:len(matrix_y[1])]))
s_sq_y3 = average([(y3j - y3)**2 for y3j in matrix_y[2][m+1:len(matrix_y[2])]))
s_sq_y4 = average([(y4j - y4)**2 for y4j in matrix_y[3][m+1:len(matrix_y[3])]))

```

```

Gp = max([s_sq_y1, s_sq_y2, s_sq_y3, s_sq_y4])/sum([s_sq_y1, s_sq_y2, s_sq_y3,
s_sq_y4])

f1 = m-1
f2 = n
f3 = f1*f2
q = 0.05

def cohren(f1, f2, q=q):
    q1 = q / f1
    fisher_value = f.ppf(q=1 - q1, dfn=f2, dfd=(f1 - 1) * f2)
    return fisher_value / (fisher_value + f1 - 1)
Gt = cohren(f1, f2)
if Gp < Gt:
    print("Дисперсія однорідна")
    print(f"Gp={round(Gp,3)} < Gt={round(Gt,3)}")
else:
    print("Дисперсія неоднорідна")

# Student criteria DONE
s_sq_aver = average([s_sq_y1, s_sq_y2, s_sq_y3, s_sq_y4])
s_sq_b = s_sq_aver/(n*m)
s_b = math.sqrt(s_sq_b)

student = partial(t.ppf, q=1 - 0.025)
t_student = student(df=f3)

bet0 = sum(list(map(lambda x, y: x*y, [y1, y2, y3, y4], [row[0] for row in
matrix_y])))/n
bet1 = sum(list(map(lambda x, y: x*y, [y1, y2, y3, y4], [row[1] for row in
matrix_y])))/n
bet2 = sum(list(map(lambda x, y: x*y, [y1, y2, y3, y4], [row[2] for row in
matrix_y])))/n
bet3 = sum(list(map(lambda x, y: x*y, [y1, y2, y3, y4], [row[3] for row in
matrix_y])))/n

t0 = abs(bet0)/s_b
t1 = abs(bet1)/s_b
t2 = abs(bet2)/s_b
t3 = abs(bet3)/s_b

ts = [t0, t1, t2, t3]
res_c = [x[1] for x in list(map(lambda t, b: [t, b], [t0, t1, t2, t3], [b0, b1,
b2, b3])) if x[0] > t_student]
res_t = [x[0] for x in list(map(lambda t, b: [t, b], [t0, t1, t2, t3], [b0, b1,
b2, b3])) if x[0] > t_student]
excluded_c = [x[1] for x in list(map(lambda t, b: [t, b], [t0, t1, t2, t3], [b0,
b1, b2, b3])) if not x[0] > t_student]

for i in range(4):
    matrix_natur[i].insert(0, 1)
def result_x(ni):
    res_x = []
    for t in res_t:
        if ts.index(t) == res_t.index(t):
            res_x.append(matrix_natur[ni][ts.index(t)])
    return res_x

print(f"Коефіцієнти {[round(c,3) for c in excluded_c]} приймаються незначними")

```

```

при рівні значимости 0.05, тобто вони виключаються із рівняння")
print("Коефіцієнти, що залишилися " + str([round(c,3) for c in res_c]))

def regression(b, x):
    return round(sum(list(map(lambda i, j: i*j, b, x))),3)

print(f"Значення 'y' із коефіцієнтами {[round(c,3) for c in res_c]}:")
for i in range(1, 4):
    print("y{0} = {1}".format(i, regression(res_c, result_x(i-1))))

# Fisher criteria
y1_f = regression(res_c, result_x(0))
y2_f = regression(res_c, result_x(1))
y3_f = regression(res_c, result_x(2))
y4_f = regression(res_c, result_x(3))
d = len(res_c)
f4 = n - d
s_sq_ad = (m/(n - d)) * sum(list(map(lambda y, yf: (y - yf)**2, [y1, y2, y3, y4], [y1_f, y2_f, y3_f, y4_f])))
fp = s_sq_ad/s_sq_b

fisher = partial(f.ppf, q=1 - 0.05)
f_t = fisher(dfn=f4, dfd=f3)
if fp < f_t:
    print(f'Fp={fp} < Ft={f_t}, отже математична модель адекватна експериментальним даним')
else:
    print(f'Fp={fp} >= Ft={f_t}, отже математична модель не адекватна експериментальним даним')

```